(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 March 2002 (21.03.2002)

PCT

(10) International Publication Number
WO 02/23900 A2

(51) International Patent Classification[7]: H04N 7/00

(21) International Application Number: PCT/US01/27253

(22) International Filing Date: 31 August 2001 (31.08.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/661,486 14 September 2000 (14.09.2000) US

(71) Applicant: AGILE TV CORPORATION [US/US]; Building 202, 333 Ravenswood Avenue, Menlo Park, CA 94025 (US).

(72) Inventors: CALDERONE, Theodore; 422 Portofino Drive #3, San Carlos, CA 94070 (US). FOSTER, Mark, J.; 4157 Interdale Way, Palo Alto, CA 94306 (US).

(74) Agents: GLENN, Michael et al.; Glenn Patent Group, Suite L, 3475 Edison Way, Menlo Park, CA 94025 (US).

(81) Designated States (*national*): AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 02/23900 A2

(54) Title: N-WAY DEMULTIPLEXER

(57) Abstract: Cable systems vary dramatically in the number of channels that they have to support. The invention provides the ability to have anywhere from one up to 96 different channels of output, while freely intermixing the number of channels that are bonded together under this output. The invention allows one to select the number of channels to be bonded together onto the output arbitrarily.

N-Way Demultiplexer

BACKGROUND OF THE INVENTION

TECHNICAL FIELD

The invention relates to data transfer within a communications system, such as a digital television distribution network. More particularly, the invention relates to a technique for mediating data exchange rates among various components of a digital television distribution network by use of a novel n-way demultiplexer.

DESCRIPTION OF THE PRIOR ART

Agile TV of Menlo Park, California has developed a system that uses an extremely powerful compute engine to perform various tasks, including speech recognition and Web browsing, see Calderone et al., *System, Method and Node of a Multi-Dimensional Plex Communication Network and Node Thereof*, U.S. Patent Application Serial No. 09/679,115, filed October 4, 2000. Due to the very high computational capabilities of the compute engine, as well as its interconnected bandwidth, a single output processor is capable of outputting a continuous data stream on the order of 2.6 gigabits per second on a single output port. The preferred compute engine may be configured with anywhere from one to sixteen output ports, although a presently preferred configuration includes two output ports.

different architectures where the distribution of the signal varies, depending on the architecture of the specific cable headend.

An output clock synchronizes an output CPU with an n-way demultiplexer to allow the demultiplexer to know which output is which. To do so, the invention provides a synchronization scheme in which a synchronization string is always written to channel zero before the output is allowed to be clocked. Once synchronization is established, each channel has its own word-length output buffer. Thus, each time the clock sends out a signal, a new word is put into the output buffer, unless it happens to be for channel zero which does not need a memory.

An address counter controls the output buffer. When the address counter is counting it is pointing to one of 95 sixteen-bit shift registers that are associated with the output buffers. For example, channel one is written with a first word, then channel two, then channel three, then channel four, and then channel five – up to channel 95. When the counter wraps around to zero, the synchronization string is expected. The address counter continues to point at zero until the synchronization string is detected. Thus, the invention provides a mechanism that automatically re-synchronizes itself. For example, in the event that something goes wrong and synchronization is lost, the invention provides a mechanism that waits for a synchronization string and that then re-synchronizes on its own.

While the data are written to the shift register, the output clock is performing a shift register function. Thus, the data are input in parallel and then shifted out in serial. On the first clock edge the zero bit is shifted out, on the next clock edge the one bit is shifted out – up through fifteen. By the time the shift register reaches fifteen and it is

Fig. 4 is a detailed block schematic diagram of the n-way demultiplexer of Fig. 2 showing an alternative, equally preferred clock selection logic circuit according to the invention; and

Fig. 5 is a flow diagram showing an example of a preferred synchronization sequence according to the invention.

DETAILED DESCRIPTION OF THE INVENTION

The invention provides a high performance, low cost method of distributing high data rate output data to a number of different ports.

Fig. 1 is block schematic diagram of the output section of a communications system according to the invention. In Fig. 1, an output CPU 11 is connected to a compute engine 12, 13 by a bridge circuit 15. Such circuits are very well known in the art. The system may comprise one or more CPU's.

This high speed bus provides 800 megabytes per second of bandwidth, thereby enabling sufficient connectivity to saturate the output port. It will be appreciated by those skilled in the art that other interconnect strategies may be used to implement the invention and that the implementation discussed herein is provided for purposes of example and not by way of limitation of the invention.

registers, *e.g.* 30, 31, 32, has a carry output which is connected to the carry input of the shift register in the next lower numbered output channel.

- Clock generation logic 36 that provides for the generation of a variety of different output clocks. Eight different clocks are provided in the preferred embodiment of the invention. Each of these eight clocks may be programmed to be an integral divisor of a reference clock 34. The reference clock may be either identical to the output clock 35 provided to the output CPU's FIFO, or it may be an integer multiple of that clock. If the reference clock is run at the CPU's output clock rate, the clock divisors may range from 1-96, enabling from one (with a divisor of 96) to 96 (with a divisor of one) output channels. If the reference clock is run at an integer multiple of the output clock associated with the output CPU, then the divisor ranges must support the same range mentioned above, multiplied by the degree b which the speed increase of the reference clock is greater than the output CPU's FIFO clock. For example, the reference clock could be run as high as sixteen times the CPU's output clock. In this case, the clock divisor ratios must range from 1 to 96 * 16.

- Clock selection logic associated with each individual output stage, which enables each output to be run at one of the eight different clock rates, independently of the other outputs. Figs. 3 and 4 provide more detailed block schematic diagrams showing alternative clock selection logic according to the invention.

In the preferred embodiment of the invention, channel 0 is reserved for the detection of synchronization information. Synchronization is necessary to ensure that the next

- If at any time channel 0 is written with any value other than 0FFFFH (140), the address counter continues to hold at 0 (150). Only after the address counter is written with 0FFFFH is the count permitted to advance to the active output channel and data are output (160).

- In the event that loss of synchronization occurs (150), FIFO data are repeatedly written to channel 0 until a value of 0FFFFH appears in the stream 120). Note that this does not inherently guarantee immediate synchronization, but it takes no more than a few loops through the counter outputs before synchronization occurs, typically within a matter of milliseconds.

In addition to the synchronization scheme discussed above, the n-way demultiplexer provides another important capability, *i.e.* the demultiplexer may be configured to support anywhere from one to 95 channels on its serial outputs in a manner which is substantially transparent to system software. Each cable system may require a different number of channels to be multiplexed onto each serial output stream. The n-way demultiplexer enables the number of channels that are bonded together to be set independently for each output channel.

To accomplish this task, at system initialization each output channel is configured to run at an appropriate clock rate. To bond channels together, it is only necessary to select the same rate for sequential output channels, and then the lowest numbered channel's output is actually used for output, while the remaining output pins are ignored. For example, if it is necessary to bond eight outputs together into a serial data stream, inputs 10-18 could all be programmed with a clock divisor (96/8=12), meaning that they are clocked at a rate which is $1/12^{th}$ that of the CPU's FIFO clock.

CLAIMS

1. A method for distributing high data rate output data to a number of different ports, comprising the steps of:

providing an output module having a plurality of different output channels;

providing a demultiplexer in communication with said output module for receiving said output channels; and

providing an output clock for synchronizing said output module with said demultiplexer.

2. The method of Claim 1, further comprising the step of:

bonding together any predetermined number of channels for each of said ports independently of each of said other ports.

3. The method of Claim 1, further comprising the step of:

providing a synchronization scheme in which a synchronization string is always written to a particular channel before said output channels are allowed to be clocked.

4. The method of Claim 3, further comprising the step of:

providing each output channel with its own output buffer;

wherein, once synchronization is established, each time said clock sends out a signal, a new word is put into an appropriate output buffer.

5. The method of Claim 4, further comprising the step of:

9. The demultiplexer of Claim 8, wherein each of said shift registers has a carry output which is connected to a carry input of a shift register in a next lower numbered output channel.

10. The demultiplexer of Claim 9, further comprising:

clock generation logic for generating a plurality of different output clocks;

wherein each of said output clocks may be programmed to be an integral divisor of a reference clock;

wherein said reference clock may be either identical to an output clock provided to said output module, or it may be an integer multiple of said clock.

11. The demultiplexer of Claim 10, further comprising:

clock selection logic associated with each individual output buffer for enabling each output buffer to be run at one of a plurality of different clock rates, independently of said other output buffers.

12. The demultiplexer of Claim 8, wherein a channel is reserved for detection of synchronization information to ensure that data transferred between said output module and said demultiplexer is written to a proper output buffer.

13. The demultiplexer of Claim 12, wherein said address counter is inhibited from advancing past a first address until said reserved channel is written with a synchronization string.

14. A synchronization method for a demultiplexer, comprising the steps of:

using a lowest numbered channel's output for actual data output, while ignoring remaining outputs.

17. An apparatus for distributing high data rate output data to a number of different ports in a system having an output module having a plurality of different output channels, said apparatus comprising:

a demultiplexer in communication with said output module for receiving said output channels; and

an output clock for synchronizing said output module with said demultiplexer.

18. The apparatus of Claim 17, further comprising:

means for bonding together any predetermined number of channels for each of said ports independently of each of said other ports.

19. The apparatus of Claim 17, further comprising:

a synchronization scheme in which a synchronization string is always written to a particular channel before said output channels are allowed to be clocked.

20. The apparatus of Claim 19, further comprising:

an output buffer each output channel;

wherein, once synchronization is established, each time said clock sends out a signal, a new word is put into an appropriate output buffer.

21. The apparatus of Claim 20, further comprising:

an address counter for controlling each said output buffer;

25. The method of Claim 24, wherein each of said shift registers has a carry output which is connected to a carry input of a shift register in a next lower numbered output channel.

26. The method of Claim 25, further comprising the step of:

generating a plurality of different output clocks with clock generation logic;

wherein each of said output clocks may be programmed to be an integral divisor of a reference clock;

wherein said reference clock may be either identical to an output clock provided to said output module, or it may be an integer multiple of said clock.

27. The method of Claim 26, further comprising the step of:

enabling each output buffer to be run at one of a plurality of different clock rates, independently of said other output buffers with clock selection logic associated with each individual output buffer.

28. The method of Claim 24, wherein a channel is reserved for detection of synchronization information to ensure that data transferred between said output module and said demultiplexer is written to a proper output buffer.

29. The method of Claim 28, wherein said address counter is inhibited from advancing past a first address until said reserved channel is written with a synchronization string.

means for configuring each output channel to run at an appropriate clock rate at system initialization;

means for selecting a same rate for sequential output channels; and

means for using a lowest numbered channel's output for actual data output, while ignoring remaining outputs.

This Page Blank (uspto)

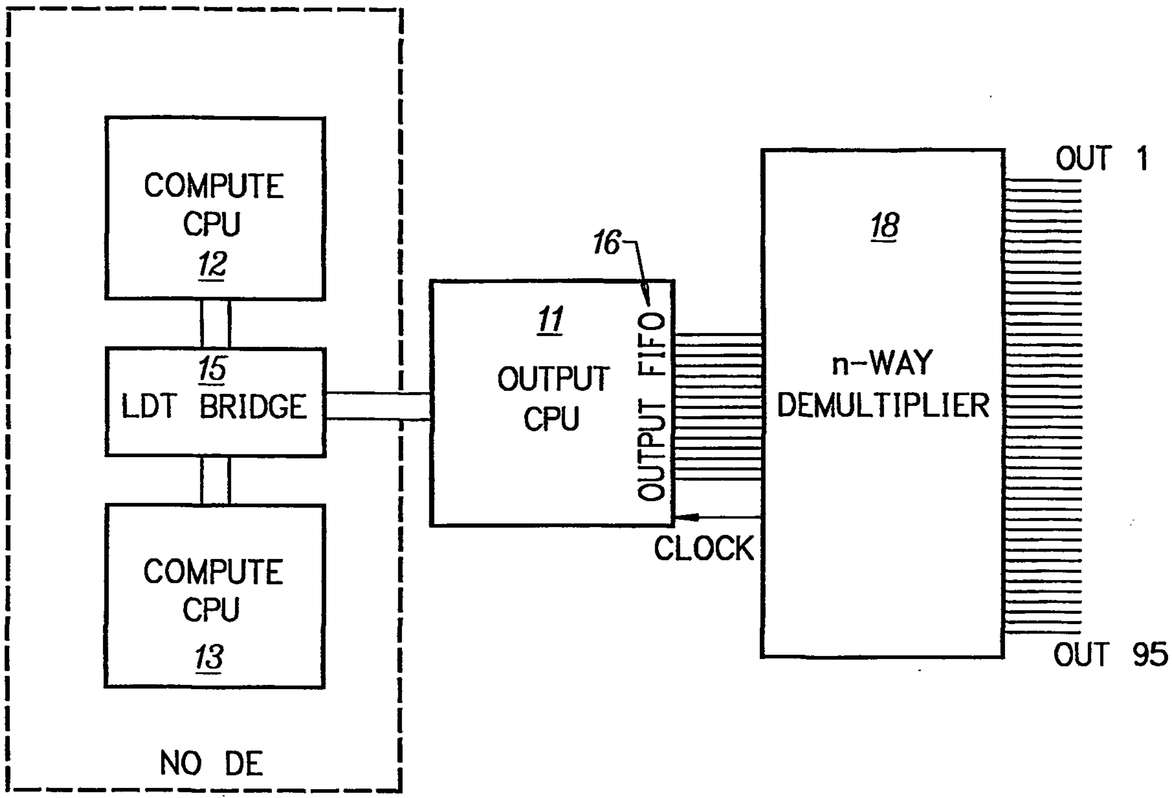
COMPUTE
CPU
12
15
LDT BRIDGE
COMPUTE
CPU
13
NO DE
16
11
OUTPUT
CPU
OUTPUT FIFO
CLOCK
18
n-WAY
DEMULTIPLIER
OUT 1
OUT 95

FIG. 1

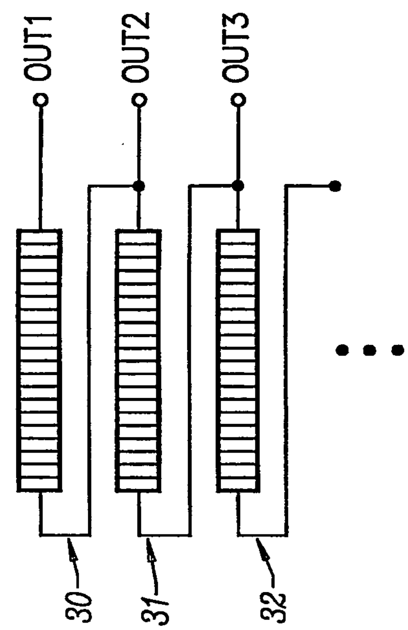
OUT1
OUT2
OUT3
30
31
32

FIG. 2A

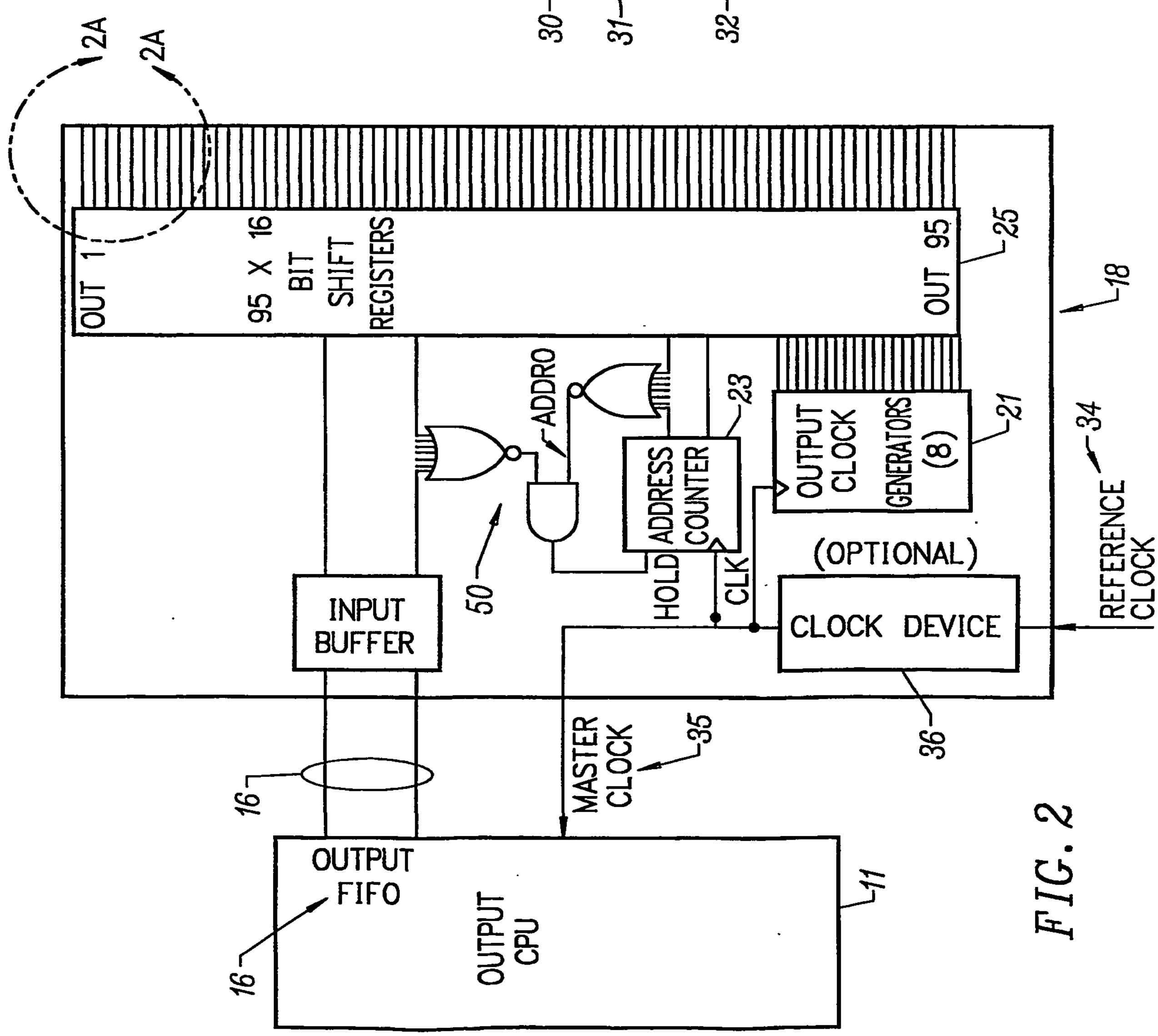
2A
2A
OUT 1
95 X 16
BIT
SHIFT
REGISTERS
OUT 95
25
18
ADDRO
23
ADDRESS
COUNTER
HOLD
CLK
OUTPUT
CLOCK
GENERATORS
(8)
21
34
REFERENCE
CLOCK
(OPTIONAL)
CLOCK DEVICE
36
INPUT
BUFFER
50
MASTER
CLOCK
35
16
OUTPUT
FIFO
16
OUTPUT
CPU
11

FIG. 2

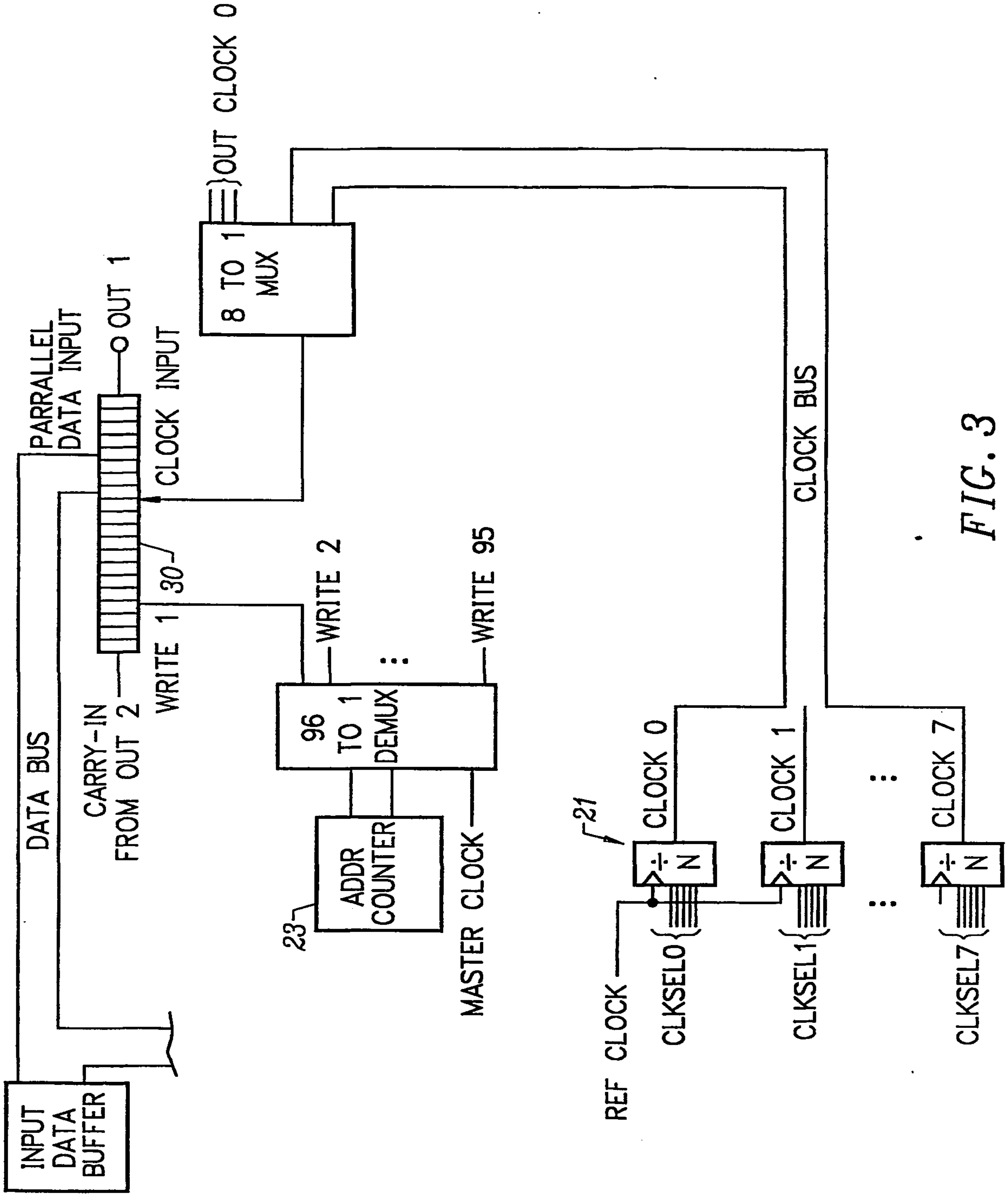
OUT CLOCK 0
8 TO 1 MUX
PARRALLEL DATA INPUT
OUT 1
CLOCK INPUT
30
WRITE 1
WRITE 2
WRITE 95
CARRY-IN FROM OUT 2
DATA BUS
96 TO 1 DEMUX
ADDR COUNTER
23
MASTER CLOCK
INPUT DATA BUFFER
CLOCK BUS
CLOCK 0
CLOCK 1
CLOCK 7
21
÷ N
REF CLOCK
CLKSEL0
CLKSEL1
CLKSEL7

FIG. 3

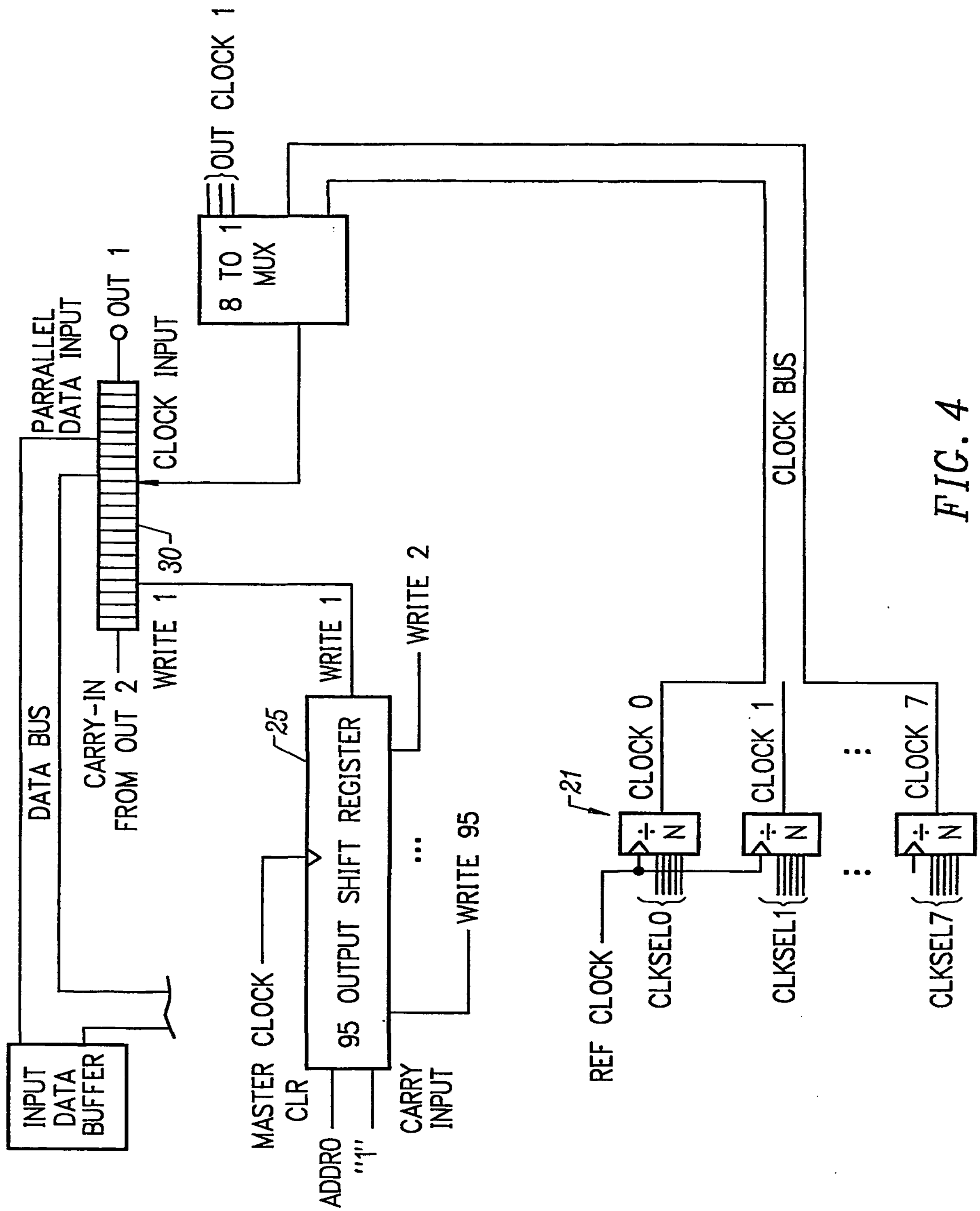
INPUT DATA BUFFER
DATA BUS
PARRALLEL DATA INPUT
CARRY-IN FROM OUT 2
WRITE 1
30
CLOCK INPUT
OUT 1
8 TO 1 MUX
OUT CLOCK 1
WRITE 1
WRITE 2
25
95 OUTPUT SHIFT REGISTER
WRITE 95
MASTER CLOCK
MASTER CLR
ADDR0
"1"
CARRY INPUT
CLOCK BUS
21
CLOCK 0
CLOCK 1
CLOCK 7
÷ N
÷ N
÷ N
REF CLOCK
CLKSEL0
CLKSEL1
CLKSEL7

FIG. 4

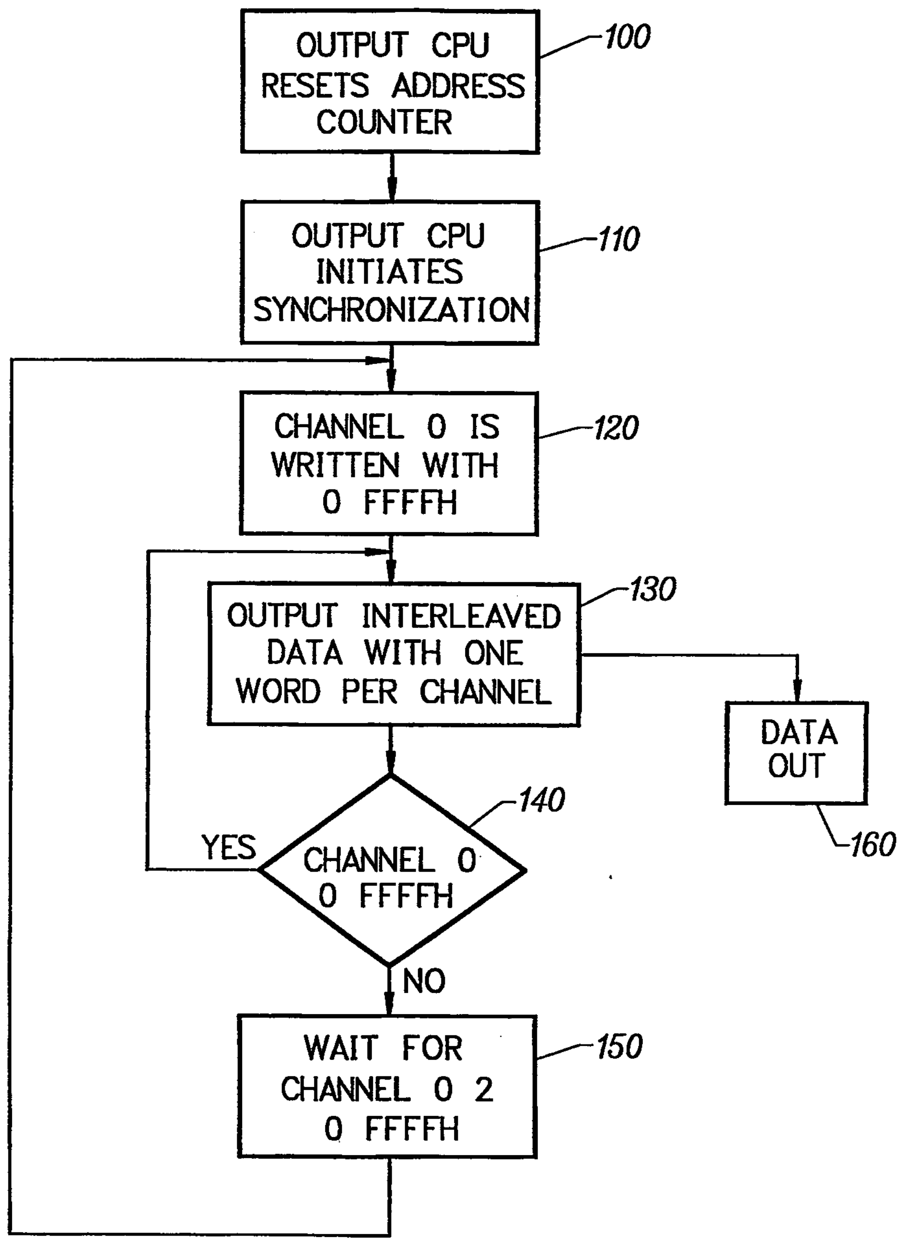
OUTPUT CPU RESETS ADDRESS COUNTER
100
OUTPUT CPU INITIATES SYNCHRONIZATION
110
CHANNEL 0 IS WRITTEN WITH 0 FFFFH
120
OUTPUT INTERLEAVED DATA WITH ONE WORD PER CHANNEL
130
DATA OUT
160
CHANNEL 0 0 FFFFH
140
YES
NO
WAIT FOR CHANNEL 0 2 0 FFFFH
150

FIG. 5